Quadruped Animation

Ljiljana Skrba, Lionel Reveret, Franck Hétroy, Marie-Paule Cani, Carol O'Sullivan

Quadruped Animation

Ljiljana Skrba[1], Lionel Reveret[2], Franck Hétroy[2], Marie-Paule Cani[2] and Carol O'Sullivan[1]

[1]Graphics, Vision and Visualisation group (Trinity College Dublin)
[2]EVASION - LJK (CNRS, INRIA and Univ. Grenoble)

Abstract

Films like Shrek, Madagascar, The Chronicles of Narnia and Charlotte's web all have something in common: realistic quadruped animations. While the animation of animals has been popular for a long time, the technical challenges associated with creating highly realistic, computer generated creatures have been receiving increasing attention recently. The entertainment, education and medical industries have increased the demand for simulation of realistic animals in the computer graphics area. In order to achieve this, several challenges need to be overcome: gathering and processing data that embodies the natural motion of an animal – which is made more difficult by the fact that most animals cannot be easily motion-captured; build accurate kinematic models for animals, in particular with adapted animation skeletons; and develop either kinematic or physically-based animation methods, either embedding some a priori knowledge about the way that quadrupeds locomote and/or building on some example of real motion.

In this state of the art report, we present an overview of the common techniques used to date for realistic quadruped animation. This includes an outline of the various ways that realistic quadruped motion can be achieved, through video-based acquisition, physics based models, inverse kinematics, or some combination of the above. The research presented represents a cross fertilisation of vision, graphics and interaction methods.

Categories and Subject Descriptors (according to ACM CCS): Quadruped Animation, Behavioural Simulation

1. Introduction

Aslan in the film The Chronicles of Narnia looks and acts like a real lion [HDK*06]. He has changed our expectations of how well we can represent animations of furry creatures. In addition, there has been some recent commercial work on realistic simulation of large herds of animals as can be seen in games such as Afrika for PlayStation3®. The game features replications of much of Africa's flora and fauna, where large herds of wildebeest and zebras can be displayed in real-time. Over the years, much research has been concentrated on the simulation of realistic humans and there are many courses and surveys relating to this topic (e.g., [TT04, BK05, WBK*07]). However, as can be seen from the examples above, others have also been active in developing realistic simulation methods for a variety of non-human characters, particularly for quadrupeds.

With respect to traditional animation techniques, standard authoring tools are now available for animating quadrupeds. However, as with all animations that are created using key-frame animation, the realism of the final motion depends on the knowledge and skill of the animator. Procedural quadruped animation allows the automation of this task, but has the drawback that the animator has less direct control over the motions, which can result in a loss of realism [vdP96]. Difficulties arise when trying to animate complex articulated structures, with the compounding factor that humans are very familiar with such motions and can detect anomalies quite easily. Motion capture can provide large amounts of detailed data that can be used to replicate the motion of a character's performance with a high level of fidelity. Such data is hard to come by for animals, however, due to a number of drawbacks. Sturman provides a short history of the use of motion capture for computer character animation in [Stu94]. While animal motion capture may not always be feasible, there have been some recent successes with using video-based methods to extract motion data, described in section 2.2. However, it should be noted that motion capture can only model one, given existing motion; the

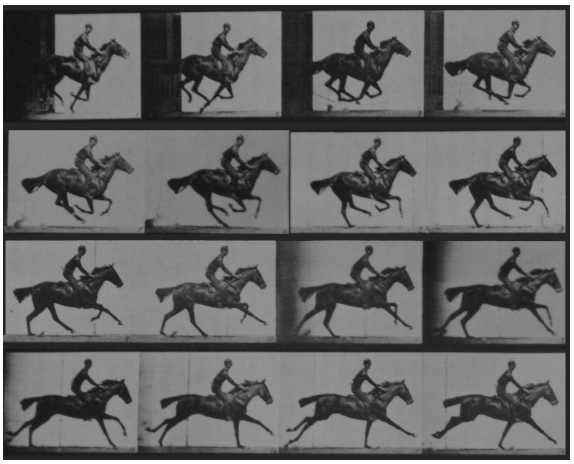

Figure 1: *Sequence of pictures capturing the motion of horse and rider. (Muybridge [EM57])*

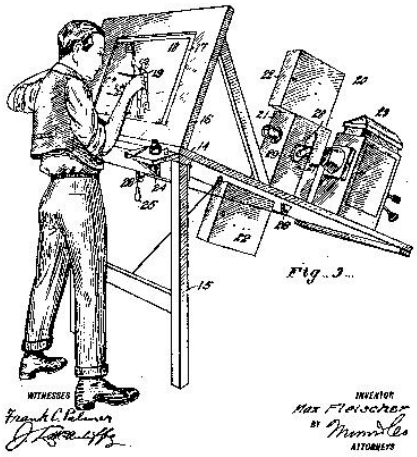

Figure 2: *Rotoscoping by Max Fleischer. (Fleischer)*

user will need to edit it, parameterize it and adapt it to their purposes; very often, they will simply want to learn from or be inspired by it, then apply what they have learned to other animals with varying morphologies, perhaps locomoting on another terrain, with a different trajectory and combination of successive gaits.

In this report, we aim to bring together and review recent research which has contributed to the creation of realistically animated four-legged characters. We will examine techniques that allow us to gain a better understanding of the movements of such animals' limbs and also the different ways in which realistic motions can be recreated.

1.1. Optical recording of quadruped motion

The biomechanics and animal physiology literature provides several sources of data upon which the synthesis of realistic motion for quadrupeds [EM57, EKB95] is based. In the late 1800s the works on motion photography of Eadweard Muybridge in the US and Etienne-Jules Marey in France greatly improved the understanding of animal gaits. Using a trigger, Eadweard Muybridge was able to set off a series of twenty-four cameras in order to capture horses, deer, buffaloes, camels, dogs, cats and many more animals galloping. Pictures of these animals represent only a subset of the 3,919 photographs shown in [EM57]. Etienne-Jules Marey used a different technique called chronophotography, where several phases of a single moment could be recorded on the same image. Images of animals in motion can show, for example, that there is a suspension phase during a horse gallop where all the hooves are above the ground but under the body as can be seen in the top row of Fig. 1. Still images such as these are still valuable sources of motion even today, especially in situations where hand animation is used for animated feature films, video games and entertainment applications.

Rotoscoping is a technique which was developed in the early 1900s and is still widely used today in animation and for special effects. The basic process consists of a video projected one frame at a time onto a glass screen. A sheet of translucent paper is placed on top of the glass and the image of the video frame is traced onto the sheet, see Fig. 2. A different sheet is used for each frame. The images are then re-photographed onto the cinematic film using a rotoscope machine [Mil98]. This results in character motion that is very smooth and life-like, which is very difficult to achieve by hand animation alone. The results of this technique can be readily found in many Disney feature animations, such as Snow White, Fantasia, and Mary Poppins. The most relevant movies to us are Lady and the Tramp, Bambi and The Jungle Book, where frequent trips to nearby farms and zoos took place in order to film the animals [TJ95]. Similarly, in the movie 101 Dalmatians, rotoscoping was mainly used for animating human and animal motion.

Today the same principles apply, except that computers are used to digitally scan the film onto disk. Then, specialized software is used to apply stylisations to the video (e.g., Shake, FFI and Pinnacle Commotion) [Sil04]. Rotoscoping can even be thought of as an alternative to motion capture. The motion of the actors is captured by hand and, as already mentioned, is often used to capture and recreate the motions of four-legged creatures. However, this is a very tedious and time-consuming process, as the animator must work on one frame at a time. To overcome this problem, computer vision solutions have been found for tracking contours and extracting animal motion from video footage in an efficient way [AHSS04, FRDC04]. This is discussed in detail later in the paper.

1.2. Biomechanics of animal motion

Quadruped motion has long been an active area of research in biomechanics and zoology, which can provide valuable insights into generating automatic gaits for various quadruped characters. Alexander has published numerous reference works on animal motion [Ale68, AJ83, Ale84, Ale96, Ale03]. In particular, he developed a hypothesis about the relationship between size, speed, mass and external forces. The resulting *dynamic similarity hypothesis* states that the movements of two bodies can be described as dynamically similar if the motion of one can be made identical to that of the other by multiplying a) all linear dimensions by some constant factor, b) all time intervals by another constant factor, and c) all forces by a third factor. Under the assumption that the weight is the dominant external force involved in locomotion, the dynamic similarity hypothesis shows that motions dynamically equivalent have a constant quantity v^2/gh, where v is speed, h is the height of the hip from the ground in normal standing position and g is the gravitational acceleration. This quantity is known as a *Froude number*.

The consequence of this hypothesis is that any animal, whatever his size and weight, tends to change gaits at equal Froude numbers. This rule predicts that species change from a symmetric gait to an asymmetric gait when their Froude number is between 2 and 3. This has been verified experimentally by Alexander and Jayes on wide range of animals, from rodents to rhinoceros [AJ83]. It can can be used as a guide for animating models of different shapes and sizes by providing guidelines on how fast an animal should be moving depending on its size and what gait should be used. Keeping the relative gait speeds consistent in an environment of varying animals should serve to increase the realism of the simulation.

1.3. Animal robotics

Physical reconstructions can also provide great insights into the motion of quadrupeds for such things as balancing, managing difficult terrain, changing gaits and steering. Examples of such robots are "BigDog" and "Little Dog" built by Boston Dynamics [Bos]. BigDog's legs are shaped like that of an animal which can absorb shocks and recycle energy as it moves from one step to the next. Similarly LittleDog is used for the study of locomotion. A more familiar example is that of the Sony AIBO robot. Shaped like a dog, the robot can be programmed and used for vision studies (so that the robot can "see") and gait studies (so the robot can move around while keeping balance) [GLH04]. However all these machines still move like robots. BigDog's legs are synchronised differently to the more common quadruped gaits (the diagonal legs move at the same time), whereas LittleDog on the other hand has the typical quadruped motion (for example: front left, back right, front right, back left) but the motion is still very jerky and slow as it tackles rough terrain.

We have just given a short historical overview of research on the motion of animals in different fields. The rest of the paper is broken down as follows: Section 2 discusses the different methods used to gather the data needed to create data drived quadruped animation. Section 3 deals with the actual animation of a character. This covers the animation of quadrupeds, with skeletons using inverse kinematics, or without a skeleton using mesh deformation. Section 4 covers the work that has been done on physical simulations of animals, the creation of anatomically correct creatures and controlling the behavior of animals through user interfaces.

2. Data driven approaches

One approach to generating quadruped motion is to capture and apply the actual motions of real animals. This may be as simple as visually observing their motion, or as complex as using multiple cameras for full-body motion capture. Reliable and realistic results, however, can be difficult to achieve.

2.1. Standard motion capture

With respect to the motion capture of animal motion, the first challenge is in actually attaching equipment or markers to animals in order to track their movement. Afterwards, the performance of an action must be restricted to the space covered by the cameras or magneto fields. In some cases treadmills can be used, for example with horses and dogs. However, this method can produce uncharacteristic motion - walking on grass is different to walking on a treadmill, and it is unsuitable when working with wild animals. Furthermore, any data captured will be very specific to a particular animal and its movement, making it difficult to change or blend with other movements.

In animal biomechanical studies, force plates are often used to track the force exerted by particular limbs in animals, thus determining which are the weight bearing legs. For example, in the case of dogs, the front legs act as the main support for body weight (ribs, head and other parts all rest on the front legs) while the back legs are used as the pushing force [CDNDMN85]. However, this method does not provide information about the movement of the spine, neck and head. Due to an increase in demand for computer generated animals in the movie industry, motion capture technology has been used to capture and analyse the gait of larger animals such as horses. However, this is expensive and done by specialised companies such as [Law] and [Wid]. Motion capture is also used in veterinary sciences for measuring the kinematics of horses' joints in order to better understand their movement and the origin and causes of dysfunctions which occur among horses used in competitive sports [BPB06].

Equine locomotion is a very active area of research, with dedicated conferences and journals [Cla97, Cla98, Cla01, ICE], so it is no surprise that motion capture methods have been applied to horses more than any other animals. It also

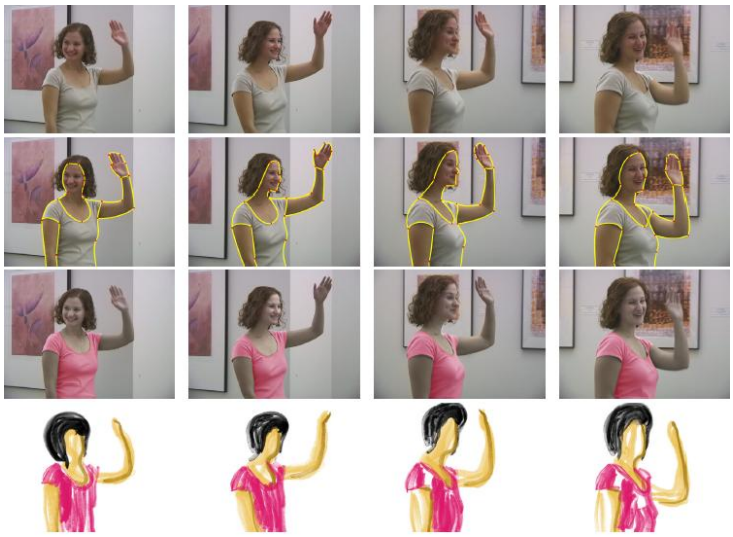

Figure 3: *Roto-curves are used to outline areas of interest (Agarwala et al. [AHSS04])*

helps that they are tame and easily-trained beasts. However, this captured motion is very specific to horses and thus not applicable to other animals. The difficulties with reuse and retargeting of such data, together with the cost and effort involved in its capture, means that motion capture for animals currently remains largely within the remit of specialised companies.

2.2. Capture from video

There are clear difficulties that arise when working with wild animals such as elephants, cheetahs, tigers and lions. Traditional motion capture is clearly unsuitable, so video processing is the most obvious solution. While there are numerous wildlife documentary videos featuring such animals, the main obstacle to exploiting this footage is the single viewpoint, making it impossible for a standard 3D measurement of motion. To overcome these problems, two types of approaches have been taken: standard tracking (e.g., [AHSS04]) and statistical analysis (e.g., [FRDC04]).

Wilhelms and Van Gelder [WVG03] use a technique to extract the motion of a horse from video, which is then applied to their three dimensional models of horses. The video image is processed using active contours. The model is scaled to match the animal in the video and it is aligned with the video. The active contours of the video are then anchored to the model of the horse. Playing the video changes the shape of the contour lines which in turn change the positions of the horse's limbs, as can be seen in Fig. 4. This method is very sensitive to noise so the user has to reinitialise active contours every few frames.

Following the same approach but with a stronger focus on user interface, Agarwala et al. [AHSS04] introduce "roto-curves" (similar to active contours) to outline areas of interest, which are specified by a user for the first and last frames of the video. For all in-between frames the curves can be calculated automatically and even be corrected by the user at any point in time, such as in cases of complex motion. For example, a person walking and waving their hand can be outlined with roto-curves. The results can be used for both special effects and animation. Different effects, like stylistic filters, can be applied to areas outlined by the roto-curves. To create cartoon style animation, user-drawn strokes can be attached to the roto-curves which are then pulled into place according to the shape of those curves, see Fig. 3.

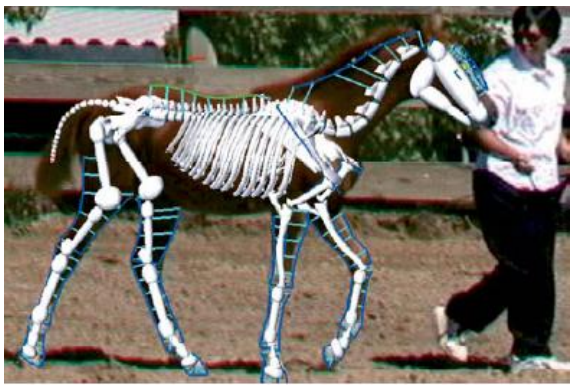

Figure 4: *Horse model is the same size as the one in the video. The active contours are anchored to the model so it follows the horse in the video. (Wilhelms and VanGelder. [WVG03])*

Gibson et al. [GODC05] present a system that captures the motions of very small scale creatures, such as spiders and ants. Three cameras are used to film the movements and then tracking techniques capture specific points on the animals. While capturing the animals, their size, the speed of their movement and the lighting in the room (diffuse lighting reduces harsh shadows) must all be taken into account. Spiders are slow moving, so cameras set at 25fps are used, whereas ants are very fast and require a frame rate increase to 150fps. For the points found in the primary view, corresponding points are found in the other two camera views, thus creating a 3D point. In order for this to work even when there is occlusion, 2D motion histograms are used to determine whether 3D points from corresponding pairs predict the expected 2D frame to frame motion. Due to the size of the creatures, very little video footage is used as input data for characteristic motion. However, the use of motion synthesis algorithms allows for the generation of large amounts of new realistic motions. This work has been used for a Natural World production by BBC's Natural History Unit.

Statistical analysis is used by Favreau et al. [FRDC04] to analyze video data and make it applicable to the generation of 3D motion. A live video sequence of a wildlife documentary is segmented into binary images in order to isolate the foreground subject from the background. This can be done automatically by image segmentation or from a rough user sketch of the parts of the animal when the automatic segmentation fails. The input visual data do not have to include detailed joints, just the main features. Principal Component Analysis (PCA) is applied directly on all the segmented images of the video sequence, each image being considered as a single high dimensional observation vector. Using PCA directly on images has been firstly proposed by Turk and Pentland [TP91] for facial images recognition. In [FRDC04], it is used to find regular motion patterns in the images and serves as an input parameter to continuously and robustly predict 3D motion of animals using Radial Basis Functions (RBF). In cases where the profile silhouettes of an animal are ambiguous, as can be the case in symmetric walks, a user is prompted to validate the silhouette as a different pose. The final collection of images is used to provide information for controlling 3D animation, as seen in Fig. 5.

Gibson et al. [GCT03] and more recently by Hannuna et al. [HCG05] have also applied PCA on visual cues from video footage of animals in motion. Gibson et al. work with dynamic outdoor scenes of walking animals. Videos of wildlife film footage can be large and contain a vast range of content (anything from large sky regions to complex foliage). They focus on recognising animal movement within such videos. The background image tends to move in a consistent manner so it is extracted leaving the foreground as the point of interest. This method works even when the camera is moving, assuming it moves in a relatively consistent manner. An eigengait model enables spatio-temporal localisation of walking animals. The eigengait space is generated using a video of a horse walking on a treadmill. Each frame is hand labelled with 12 points which are tracked over the video sequence. Using PCA analysis on all points found in the video sequence, an eigengait space is generated. Gait frequency in the outdoor scene videos is detected by tracking points found in the foreground and using their vertical differences. If the gait frequency is in the range specified by the eigengait space then the presence of an animal in the video is assumed.

Hannuna et al. [HCG05] also propose a method to detect animal movement in wildlife videos. Foreground points are identified and a bounding box is fitted over them, which specifies the dense flow area. Once again PCA is applied to a set of dense flows which describe movements. Projection coefficient variation reflects changes in the velocity and the relative alignment of the components of the foreground object. To understand the movement, motion flow images are constructed where different colors represent the direction of motion. The dense flows are able to describe the internal motion of the object's torso and not just the legs. As the motion detection is very sensitive in order to pick up subtle torso movements, other movements are also detected such as imperfect bounding box placements, head and tail movements. Furthermore, also the animals tend to move along uneven terrain with varying velocity, which detracts from the correctness of the motion flow with respect to what is being tracked. In order to improve the results, motion frequencies outside a certain range (which encapsulates gaits ranging from slow motion to galloping) are discarded as drift or noise. Results show that quadruped gait patterns are detected even in low quality videos. Using this system, movements were detected in videos which were previously missed when scanned by eye.

Ramanan and Forsyth [RF03] build 2D models of animals appearance from a video sequence. The animals are presented as a kinematic chain of rectangular segments. They build an appearance model from video footage by identifying the main pool of pixels corresponding to the animal in each frame. Once the appearance of a specific animal can be recognised by the system, the result can be used to find such animals in a collection of images. Calic et al. [CCC*05] address the problem of animal recognition and classification from videos. However, as with the models of [GCT03] and [HCG05] they only deal with the processing of videos, finding patterns in the individual frames and classifying them, rather than actually tracking the motion of the animals.

In summary, computer vision techniques have proven effective in the detection and extraction of animal data from wildlife videos. However, the data is not always suitable for use in the simulation of 3D animal characters. The approaches of Favreau et al. and Wilhelms and Van Gelder are the only cases reviewed, where the authors have successfully managed to apply some of the motions captured by video to 3D models.

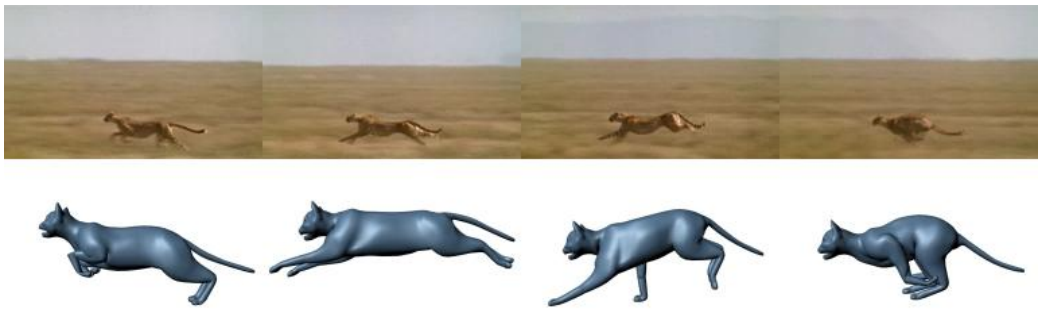

Figure 5: *Images extracted from a video are used as learning set for prediction of continuous 3D motion (Favreau et al. [FRDC04])*

3. Animation methods

Character animation methods fall into many different categories including manual, procedural, physically based or inverse-kinematics approaches. In practice, a combination of techniques is used and the amount of user interaction can vary. We now provide an overview of animation methods used to date for quadruped animation.

3.1. Creation of the animation skeleton

Animation skeletons (sometimes called I.K. skeletons, I.K. standing for "inverse kinematics") are the standard control structure to animate 3D character models. They consist of a simplified version of the true anatomical skeleton, with only the main joints being represented: an animation skeleton is defined as a hierarchy of local reference frames, each frame corresponding to a joint. Pairs (parent, child) of frames are called the "bones" of the skeleton. The appeal of using skeletons comes from the fact that they provide a very natural and intuitive interface for animating characters. For the final rendering, it is possible to deform in real-time a corresponding 3D mesh (corresponding to the skin of the character) according to its pose using standard linear blend skinning algorithm. Animation skeletons are usually created with respect to such a target 3D mesh to be animated. However, one still needs a good understanding of anatomy in order to produce adequate skeletal structures, especially for quadrupeds.

In the literature, a lot of methods have been proposed that try to guess an animation skeleton only from the geometry of the skin, represented as a mesh [KT03, LKA06, WML*06] or a set of voxels [WP02]. However, since the anatomy of a character is not always directly linked to its shape (see Fig. 6: horse's spine should be along the top of the back), these skeletons cannot be used as such for animation: a user's input is usually required to adjust them.

A few recent methods have been proposed that insert a priori anatomy knowledge in order to create animation skeletons which are more related to the real skeletal structure of the animals. This knowledge can either be inferred from a

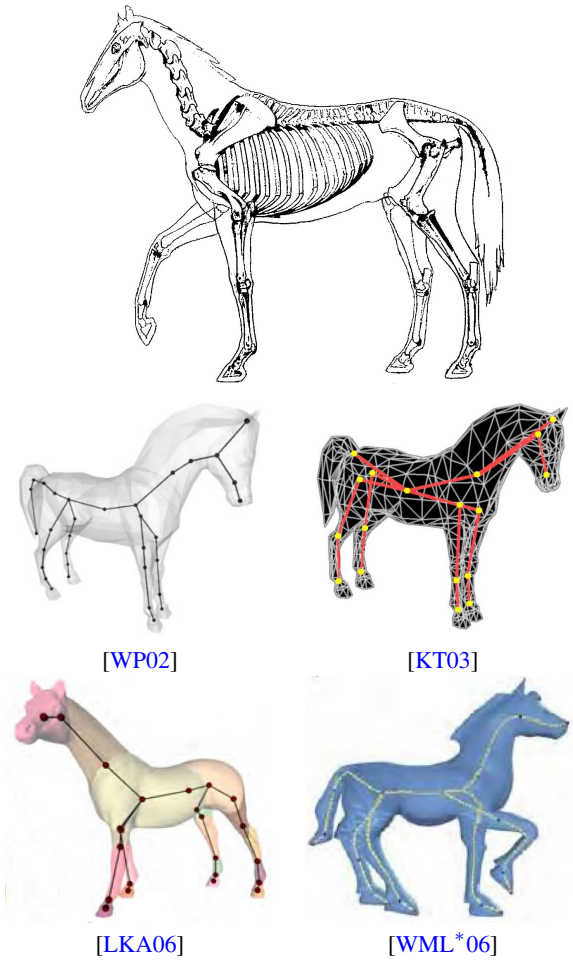

Figure 6: *Top: skeletal structure of a horse. Middle and bottom: animation skeletons inferred from the geometry of the shape with different methods.*

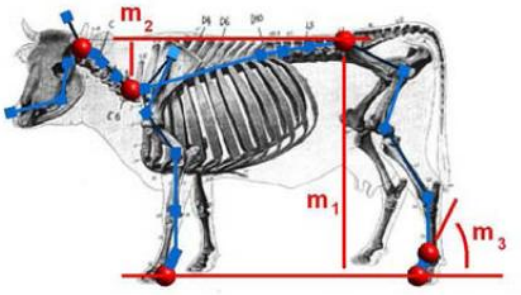

Figure 7: *Three parameters controlling the morphable model (Revéret et al. [RFDC05]).*

set of examples [RFDC05, SY07], or given as a template [MDMT*04, AHLD07, BP07].

In order to quantify morphological variation of anatomically plausible animation skeleton of quadrupeds, Revéret et al. [RFDC05] developed an intuitive method of morphing quadruped skeletons using only a few parameters, and are thus able to create skeletons for animals that are not in an initial database. Statistical analysis is applied to the collection of animation skeletons including a horse, goat, bear, lion, rat, elephant, cow, dog and pig. All the skeletons share the same topology in terms of number of articulations and joint hierarchy. Each skeleton contains information about the position and orientation of each articulation. All the skeletons are normalized to that each one has the pelvis in the same location, and the spine column is the same length for every animal. Therefore, the variability in skeletal structure between animals can be explored independently of the size of the animal. Using PCA, the authors show that using global frame coordinates and quaternion rotation, the translation and rotation data of the different models can be efficiently described by a linear model controlled by three parameters. These three parameters correspond to m_1: animal height (height from the hip to the ground); m_2: bending of the spine; and m_3: hoofed vs plantigrade, which are the three parameters used to define the shape of a geometrical model as can be seen in Fig. 7. Geometrically-controlled parameters are given more preference as they are easier to change and intuitive to use. The results show that the morphable skeleton can be fitted to any quadruped model using the three measurements mentioned above. Additionally, convincing animations can be achieved when combining the skeleton with standard smooth skinning for geometry attachment.

Rather than a database of input models and skeletons, Schaefer and Yuksel propose to use a set of input poses in order to compute an animation skeleton for a given model [SY07]. Their idea is to cluster the faces of the skin mesh into regions corresponding to rigid bones. Of course, the set of input poses should contain a rotation for each desired joint of the skeleton, unless this joint will not be recovered. The bone weights for each vertex of the mesh are then estimated using the technique described in [JT05]. Finally, these weights are used to define both the connectivity of the skeleton and the joint position, hence allowing direct creation of new poses. This method is illustrated in Fig. 8. In a similar approach, de Aguiar et al. propose to compute a skeleton from an existing mesh animation (see section 3.3) [dATTS08]. In this case, the set of input poses is replaced by a greater set of meshes with additional information, corresponding to the motion of the model.

In the case of bipeds, Moccozet et al. [MDMT*04] propose to fit a template model (described in [SMT03]), which contains both skin and animation control information (including an animation skeleton), to any scanned model. The correspondence between the two models is set manually, but the authors propose some tools to assist the user, for instance for the location of characteristic points on the input data.

Aujay et al. describe a system which is able to create a skeleton for any given model [AHLD07], with anatomically plausible location for joints in the case of bipeds and quadrupeds. A single point is selected by the user on the model as the starting point for the generation of the skeleton. This should be on the head of the character, in order that the system is able to recover the anatomy of the model. A harmonic function and a Reeb graph based on this function are then computed. The harmonic function is used to filter and recover the symmetry of the character's structure, resulting in a graph with unnecessary joints. This harmonic graph is refined starting from the user specified point and is embedded into 3D space, with anatomical information attached; this is based on a priori knowledge coming from an infographist's work. For quadrupeds the generated skeletons have been compared with those created by hand, see Fig. 9. The results show that, not only visually but also when comparing the parameters from [RFDC05], the automatic method produces valid skeletons ready for animation. Moreover, the method works even when the character's poses are different. A user interface allows for more bones to be created manually, for example if more bones were needed for better control of an elephant's trunk.

Another recent paper describes a similar system called Pinocchio which embeds a skeleton into a model [BP07]. Contrary to [AHLD07], it also attaches the skin of the model to the skeleton automatically. The skeleton construction works as follows: Spheres are fitted into the character's body and a graph is constructed based on their centers and by adding edges between spheres which intersect. This graph is then used to help embed a template skeleton. The different stages can be seen in Fig. 10, in the case of a biped. To attach the skin, a heat equilibrium approach is used, where the character is treated as a heat-conducting object. To find the weights of the vertices, the temperature of a bone is kept at 1 while the others are at 0. Equilibrium tem-

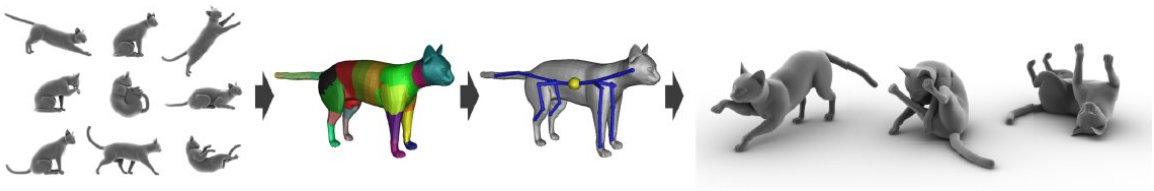

Figure 8: *A set of input poses, mesh clusters representing rigid bones, computed skeleton (the yellow dot corresponds to the root of the hierarchy), and new poses created with this skeleton (Schaefer and Yuksel [SY07]).*

Figure 9: *A cat model, its harmonic graph and the computed harmonic skeleton, along with a handmade animation skeleton in the last image (Aujay et al. [AHLD07]).*

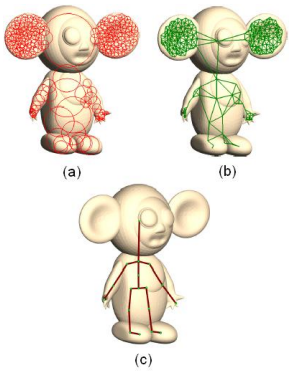

Figure 10: *(a) Spheres packed into the mesh (b) Graph constructed based on sphere intersections (c) Embedded skeleton (Baran and Popovic [BP07]).*

perature is calculated at each vertex on the surface which then specifies the weight of the bone. This results in smooth skinning, due to the second order nature of the heat equation. Note that the basic solution which bases the weights on the proximity of a bone to a vertex does not work: as the skeleton is one-dimensional, for fat characters the vertices of the belly can end up attached to the arm bones. Results show that the Pinocchio system can embed a template biped or quadruped skeleton into different models which are then ready for skeletal animation.

Skeletons that better reflect the nature of an animal should lead to better animations. On one hand, generating the skeleton automatically saves a great amount of time and does not require specific skills or anatomy knowledge from the user. On the other hand, parameterizing the skeletons provides a way of simulating new animals very quickly. It can also give better control over the levels of detail that may be needed for specific parts of a model.

3.2. Direct and inverse kinematics

One of the earliest examples of computer generated quadruped animation is the PODA computer animation system [Gir87]. The user can control the movement of the animal through a combination of inverse kinematics and dynamics simulation of the limbs and body motion. Positioning of the limbs can be achieved either through inverse kinematics by moving the end-effectors (hands and feet) to a desired position, or by rotating the joints using forward kinematics. Additionally the user can adjust the angles of the joints while keeping the end-effector in place. The different "postures" are assembled to produce the final motion path. The system interpolates between the different limb positions so that the movement is smooth. The limb trajectories are recalculated during body movement to accommodate variations in foot placement and body speed. Co-ordination is allowed, where each leg can have a state indicating whether it is in support phase (on the ground) or transfer phase (in the air). Depending on gait, the time spent in the different leg phases varies. An animal's body trajectory is solved for by taking into account its vertical motion (using Newton's equations of motion for upward velocity), horizontal motion (defined by the animator using a cublic spline) and angular motion (found by calculating the direction of movement from changes in

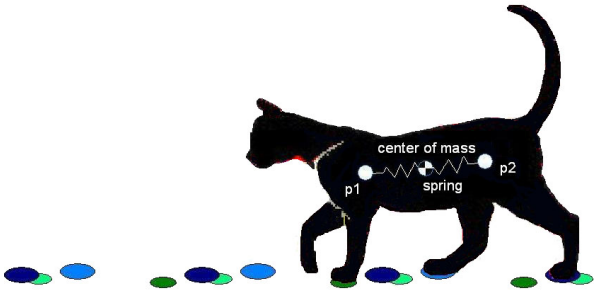

Figure 11: *Two point masses are used in the simplified physics model to make the overall body movement look realistic. The footprints guide the position of the feet, the small circles represent the front legs, and the big circles represent the back legs. (after [TvdP98])*

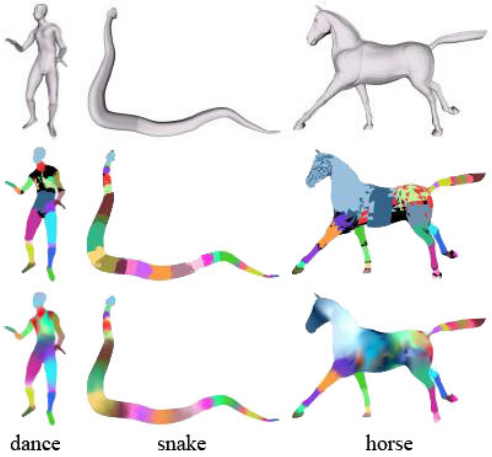

Figure 12: *(Top) the triangle mesh, (Middle) Estimated bones, (Bottom) Vertex weighting to the virtual bones (James and Twigg [JT05])*

the horizontal position and solving Newton's equations of motion). While individual limb positions are controlled by inverse kinematics, the overall movement, the timing of foot placement and body dynamics is controlled using Newton's equations in order to keep the body balanced. This can allow for better timed and more natural motion.

Torkos uses a similar approach of a hybrid physically-based/kinematic system to animate a 3-D cat [Tor97, TvdP98]. Spline trajectories represent the state of the quadruped over time. Footprint locations and their timings are additional constraints used to generate various motions such as walking, galloping, jumping, push-ups and skating for legged animals. Physics is used to make the overall movement look realistic while inverse kinematics is used for details, such as positioning the leg joints appropriately. The trajectories are used to control two key mass points. The two point masses are connected by a spring that models the internal forces of the back. The point masses are located at the hips and shoulders of the animal, see Fig. 11. The motion of the complete skeleton is reconstructed from the point mass trajectories. A trajectory is calculated from one footprint to the next. Optimisation ensures that the foot is at a reasonable distance from the body and handles collision avoidance. Inverse kinematics is used to determine the arch of the spine and the position of the leg limbs, as well as determining the movement of the head and tail. A user can control where to place the footprints on a plane, specifying the x and z co-ordinates, while y co-ordinates are calculated according to the terrain. A user interface provides suggestions about the order and timing of the feet and their positions. Additional features include automatic generation of footprints where the user specifies the path the quadruped is to follow Fig. 11. For fully autonomous behaviour, the system can generate footprints in real time, e.g, for a random wandering behaviour. To ensure that the overall movement of the 3-D cat is visually appealing, Torkos adds a "comfort" term as a constraint to make sure that the quadruped does not deviate too far from allowable positions.

Kokkevis et al. [EKB95] describe a way of creating autonomous animal motion using kinematics, dynamics and control theory. The quadruped is divided into body and legs subsystems. The body subsystem consists of position and velocity of center of gravity, orientation of the body and the angular velocity of the body. The leg subsystem consists of the position of each leg, the state of each leg (up or down) and the force it exerts of the ground. Depending on the target position and velocity of the animal, forces are combined for the body using the dynamic controller and are distributed to the legs. The leg forces are exerted on to the ground generating a reaction force on the legs, passing up to the body, thus making it move. Dynamics is used to control the legs when they are on the ground and kinematics is used to position them when they are in the air, all calculated by the gait controller, which also computes the velocity of the body and the step-size based on the gait pattern used. They successfully simulate a dog performing walking and trotting over different terrains, with minimal or no user interaction. Providing the means to generate character motions automatically allows the user to concentrate on other aspects of the animation, such as creating realistic behaviour.

While inverse kinematics is an intuitive approach to creating character animations, Torkos points out that physics can also be a useful tool for making the overall movement believable, allowing for a level of detail approach.

3.3. Mesh animation

It is also possible to create realistic animations by manipulating the skin of the mesh directly. Vertices of an articu-

Figure 13: *Skinned mesh animations (James and Twigg [JT05])*

lated mesh tend to move relative to their neighbours rather than independently. By modeling these deformations correctly, mesh manipulation is possible without the creation of a skeleton. While James and Twigg use no skeletal bones in their work, they calculate "proxy bones" that deform the skin [JT05]. The input models do not have any predefined bone transformations so proxy bone transformations are estimated before calculating vertex weights. To calculate proxy bones, triangles with similar transformations are clustered. Once the number of bones in known the vertices are weighted. The results are shown in Fig. 12. During animation a character may end up in atypical positions. Also large numbers of mesh triangles are needed for deformations. Therefore, the algorithm that assigns vertices to their respective bones has to be very robust. James and Twigg use a mean shift clustering algorithm to calculate the number of bones and the triangles associated with each bone. Each vertex can be deformed by one or more bones. To animate the model, a skinning transformation is found that maps an undeformed mesh to a sequence of deformed models. Depending on the pose of a deformed mesh, new proxy bones might be created. These are referred to as flexi-bones and they ensure smoother skin deformation. Using this method James and Twigg [JT05] were able to simulate a large herd of skinned animals (horses, camels and elephants see Fig. 13) at interactive frame rates.

Der et al. [DSP06] reduce the deformability of a model making it easier for a user to animate various creatures, and demonstrate their algorithm on a horse, elephant, mouse, gorilla, human and a dragon. By moving vertex handles created for the hooves, a horse model can be placed into a galloping pose. The deformability of the model is reduced by taking into account the fact that neighbouring vertices move together. In this way, a compact set of control parameters can be used to change the pose of the model. The control parameters allow for direct manipulation of the model. Groups of vertices that are affected by the same control parameter are replaced by a proxy vertex. This in turn speeds up the calculation of transformations for those vertices. Vertices that are affected by more than one control parameter have the influence of each parameter divided between them using a measure to weigh which control parameter has the greater influence. However, the user is still allowed to manipulate each vertex of the model. Additionally, inverse kinematics (similar to the more conventional IK algorithm for jointed rigid skeletons) is applied to the reduced model to re-position the vertices as the controllers are being moved around.

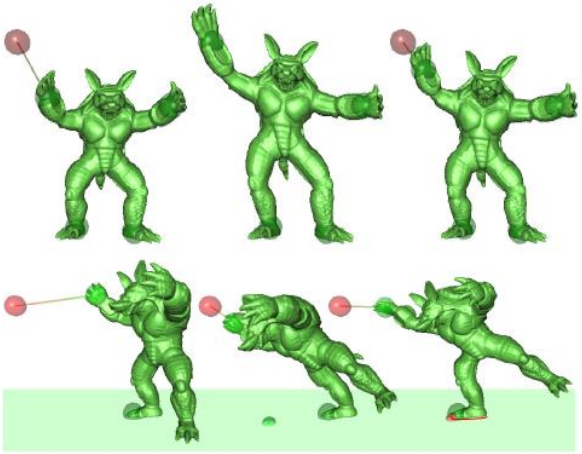

Figure 14: *(Top) When the length constraint is used, the whole body moves (right) as opposed to just dragging the arm (middle) column. (Bottom) Using the center of gravity constraint, the small red sphere shows the target position of the barycenter (Shi et al. [SZT*07])*

Shi et al. address the problem of mesh manipulation at interactive framerates [SZT*07]. New poses of 3D models such as an armadillo, camel, horse, cat or dinosaur, are created by specifying constraints on leg length preservation, fixation of the foot on the ground, balance preservation and

self-collision handling. These limitations help to maintain the volume and shape of the object. This technique combines the more traditional skeleton based rigging and mesh deformation. The input consists of a triangle mesh and an associated skeleton, where the vertices of the mesh have already been matched to the bones of the skeleton. Each vertex is defined as a linear combination of the locations where it would be if it were following the movement of the associated bone. For each bone, virtual vertices (tetravertices) are added to create a tetrahedron (tetrabone). As the tetrabone moves, so too do the tetravertices, thus capturing the motion in the process. This means that the mesh deformations can be dealt with using the mesh vertices and tetravertices alone. Laplacian constraints are used to preserve the surface details of the undeformed mesh and also to allow the user to assign a target position for a mesh vertex or a set of vertices. Tetrabones are used to keep the length of the bones constant and also to deform the skin around bones in such a way that character volume is preserved. Center of gravity is calculated by computing the barycenter of each bone when the mesh is in the original pose, and then recalculating it once the transformations are applied. By ensuring that the projection of the barycenter is on a given position on the floor, the method ensures that the position of the mesh stays balanced, see Fig. 14. In order to prevent sharp bends in joints, restrictions are placed on the ranges of the bones involved. To ensure that vertex weights are well behaved the graph Laplacian is used, resulting in smooth surfaces.

Walter et al. [WFM01] describe a technique for growing patterns on different mammalian models but their method can also be used as an animation technique. A user can segment the model into cylinders so that it is represented as a group of geometries. In order to better control the cylinder shape, a user can change its parameters, referred to as "features". The cylinders make up a hierarchy where translation and rotation is inherited from the parent cylinder but, due to the pattern generation, scaling is kept local to a specific cylinder. By changing the orientation of the limbs, one can create different poses for the models which are then used for animation. However, using this technique for animation is inadvisable since the control parameters for the cylinders are very basic and not well suited to mesh deformations. Therefore, the techniques previously described are preferable.

While mesh examples do not use skeletons explicitly, they still work with the idea of vertices being controlled by bones. However, the emphasis is put on the resulting *shape* of a pose. This can avoid unsightly kinks and bends in the skin of a character and also ensures that the volume stays the same.

3.4. Automation

The quadruped motion we are all familiar with these days is usually achieved through hand animated key frames, which are then interpolated to provide smooth motions. Commercial or freely available software contain many features to aid hand animation. However, to produce quality motions the user needs to be very skilled and even then it is still a very painstaking process. Hence, there is a great demand to automate the animation of some of the more common motions characteristic to quadrupeds. Feature animations such as Madagascar and Shrek 2 combine hand animations with some procedural techniques to aid the animators [GMC*05, GAD*07]. This idea has also been adopted by other researchers.

Kuhnel [Kuh03] used footage of horse training videos to recreate realistic animations of virtual horses and riders in Maya. Automating the movement of the horse and the rider leaves more time to animate the finer details of the scene such as arm movements and facial expressions. By simulating a horse performing different gaits, the motion of the rider can be automatically generated (balancing, bouncing, leaning forward or shifting their center of gravity). To determine whether the simulated motions of the horse and the rider are correct, they are displayed alongside reference footage and a mathematical model of the movement is iteratively modified to better reflect the original video. These mathematical comparisons thereby create a base set of rules, supporting a predictive model of what will happen to a rider in any new situation. In order to achieve this, the rider and horse are rigged in Maya. The horse rig includes control points for the hooves, rear rotational pivot and a mid-back pivot (which is located in the center of the spine). However, the authors are focused on the generation of the rider's movements, other control points are used to calculate the movements of their hips, legs, spine and arms.

Francik and Trybicka-Francik [FTF03] created a system called KINE+ that imports model data (e.g., skeleton information for bipeds or quadrupeds) from either 3D Studio Max or Maya software. The workflow pipeline includes modeling the object in 3D Studio Max or Maya (this includes adding a skeleton to the object), exporting the skeleton of the model into the KINE+ system where it is animated using C++. The animation data is stored in a file which is then imported by 3D Studio Max or Maya and rendered (skinning is not supported by KINE+ so it has to be done in other software packages). The KINE+ system is based on the Discreet Character Studio package and so the bone hierarchy is the same. To help with the animation of a character KINE+ supports the following features: forward kinematics, inverse kinematics, collision detection and avoidance, key frame managment and exporting of the animation. One thing to note is that commercial software already provides its own scripting language which can be used for animating characters within a scene without the need for any exporting/importing.

Above we have discussed the different approaches that can be used for animating quadrupeds. While skeleton manipulation is the most popular method for creating such motion, some researchers have found that equally good results can also be achieved using mesh manipulation. By parame-

terising the motion and the skeletons it is possible to create a range of gaits suitable for different animal sizes.

4. Physically-based simulation

Physically-based animation involves simulating the effects of a combination of forces such as torque and gravity, thus resulting in body movement. Some researchers have created simulations of physics-based animal motions with varying degrees of user interaction.

4.1. Physically-based modeling of animals

There are many advantages to physics based modeling. By considering any movement is a result of forces that move a body, the gravity, balance and speed can be used as varying parameters. Additionally different characteristic motions that vary depending on a character's situation are easily achieved once physics controls are set up. Some of the latest work on procedural physics simulation can be seen in the new game Backbreaker, where American football tackles are all generated as the game is played, so they look different every time. While humans, have been the primary focus in state of the art procedural techniques, others have nevertheless been active in animal motion reconstruction.

Marsland and Lapeer [ML05] minimise the need for user input when animating a physics-based trotting horse, by using video footage to position the horse's legs correctly. The horse is modeled as a collection of connected bodies which are moved from the current state to a desired state through the application of torques. Each bone in the skeleton has a mass, center of gravity, inertia, width, depth and height. The bones are connected using hinge joints which have limits based on the results of studies of horse motion. Information about the size of the angles at the joints is taken from each frame and used to calculate the torques needed to move the physics based model into the desired position. This is done by the animation controller. A heading controller keeps the physics-based horse walking towards a target position. An angle is calculated between the current heading direction and the desired direction which is then added to the joints of the horse limbs. Anatomical modeling can also be used in order to improve the realism of an action of a character.

Wilhelms and Van Gelder [WG97, SWG02] modeled and animated animals using accurate models of bone, muscle and tissue. In [WG97] these components are represented using triangle meshes and ellipsoids. As the bones move the muscles change shape, which in turn deforms the skin during the animation. Modeling a character involves building a body hierarchy, designing individual muscles, voxelising the components and mapping the skin vertices to the nearest underlying component. The bones and the tissue are parameterised so that the same components can be used for different animals. Muscles are represented as deformable cylinders or ellipsoids that are divided into discrete parts. The origin and

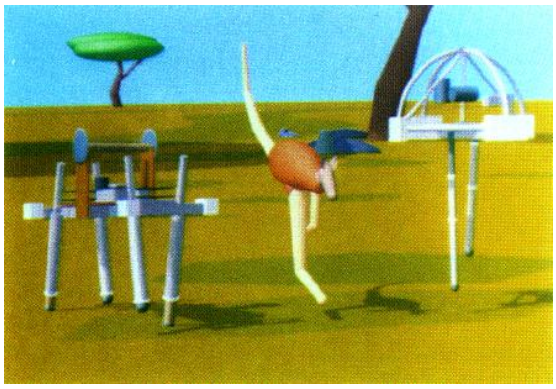

Figure 16: *Different models used in actuated systems (Rainbert and Hodgins [RH91a])*

the ends of a muscle are attached to specific bones. A user can change the shape, orientation and location of the origins and ends of a muscle. Once a joint is opened or closed, the muscle shape is recalculated. The cross section of a muscle increases as it contracts, and decreases if the muscle relaxes. The skin's vertex is attached to the closest underlying component and it can move relative to the muscle or tissue. This is because the skin is attached using virtual anchors which give the skin a more elastic appearance.

Simmons et al. [SWG02] similarly deal with the creation of 3D models of creatures, Fig. 15, where they morph a canonical representation of a character to create new creatures. The transformation is applied to the segments hierarchy (this is input into the system) and this is then propagated to the bones, muscles and skin. The input information can be gathered from either taking measurements of a real horse, or using a 3D horse model. The skin is represented by a triangle mesh. A subset of the skin vertices, designated as "feature" vertices, is used to deform the attached mesh skin. The feature vertices are created in addition to the vertices that make up the mesh of the model. Unlike in [WG97], only the feature vertices are anchored to the underlying components, which move as they move and the rest of the vertices are interpolated. This generic model can be morphed to another target animal. In order to achieve this, 89 marker locations are placed on the generic model thus denoting the points that are necessary to estimate joint location and the structure of other components such as muscle. This information is then used to transform the geometric hierarchy of the canonical model, along with its bones and muscles and skin, to match that of a target animal (which can be in a different pose). Using this technique, one is able to create very realistic animals of different sizes which can then be used in animation.

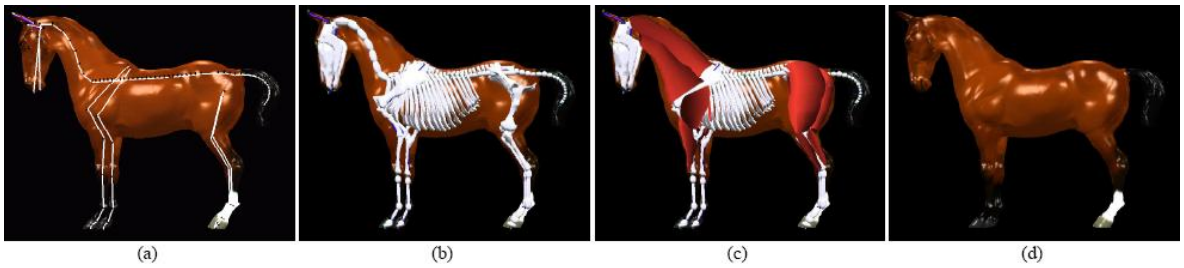

Figure 15: *(a) segment hierarchy - input into the system (b)skeleton (c)muscles (d)skin (Wilhelms and Van Gelder [SWG02])*

4.2. Designing locomotion controllers

One of the aims of computer graphics is to create systems where the user input is limited to only directing a character around a scene. The movement of the character is controlled by a system that integrates equations of motion derived from physical models, as in the work of Raibert and Hodgins [RH91a]. Users can specify the input to algorithms that control the behaviour, but they do not manipulate the model directly. Their technique was used to create a computer animated cartoon that uses automated computer characters with one, two and four legs [RH91b] (Fig. 16). The challenge here is how to simplify a high-level request so it is understood by the underlying processes while at the same time creating realistic motions. All characters in the system have control algorithms that maintain balance, hopping, speed, posture and elastic energy stored in the legs. The control inputs (speed, gait, path and the initial position of the legs) guide the actuator forces which in turn make the animal move with a desired speed and direction. The movement of the legs is modeled on a spring system. The torques calculated during the support phase are used to keep the body upright. A finite state machine is used to synchronise the legs by invoking the correct algorithm. The same technique is used for all the animals: a bipedal run is treated as a one-legged hop where the functions for a one-legged hop are applied to each leg in turn. A quadruped trotting is treated as a biped, where diagonal legs form pairs. The nice thing about this system is that, because of the higher level of control, we can ignore the individual legs.

Similarly, in the work of Laszlo et al. [LvdPF00], the details of the physically-based animations are hidden from the user. This way the user is free to experiment with the behavior of a character in the scene rather than having to control the movement of their bodies or legs explicitly. At interactive frame rates, a user can position objects around the scene and as a result can quickly produce satisfactory animations. One of the more important features of this approach is that a user can control the speed of motion through the user interface. This improves previously mentioned methods where the parameters have to be specified before the program is run. An animator is allowed to use his intuition about motions to create animations for planar characters such as lamps, cats and humans. The success of such a user interface relies on the design of effective motion controllers. For simple cases, mouse movement can be used to control the joint movement of a character where rapid mouse motion is reflected by rapid motion. This can produce jumps, shuffles and flips. For complex characters, such as a cat, keystrokes are assigned a set of desired angles assumed by the legs. By selecting different positions of legs a user can create different animations. If there is a need to correct the position assumed by the cat at any stage during the animation, checkpoints are provided that allow for more detailed positioning of the cat's limbs. The dynamics of the character are important as it means that the user will retain the necessary rhythm and timing of the action. Other complex motions like bipeds traversing monkey bars or climbing a ladder are aided by the use of IK primitives. A position that a hand is to grasp is marked and from there inverse kinematics is used to establish the desired joint angles. In addition, a finite state machine chooses the next arm that is to grasp a target. From the above we can see that, if the user interface is intuitive enough to use, and the underlying physics is abstracted appropriately from the user, many different effects can be achieved using different characters.

Ideally a motion controller should be able to control the overall movement of a character by enabling it to perform a single motion in different ways, e.g., a walking simulation that can display varying speeds and turning as well as realistic movements over uneven terrain. Different applications require different styles of animations: feature films require tightly-scripted motion control while computer games require intelligent autonomous motion. Simulated creatures capable of autonomous behaviours potentially require less effort to animate and are reusable. This involves the varied motions listed above and it also requires planning for a given task. For example, given a task such as "walk to the door", a character needs information about the environment in order to plan its route. To achieve such high level character control, one needs a rich family of well-studied control techniques but that are easy to use.

Van de Panne provides an overview of methods used for

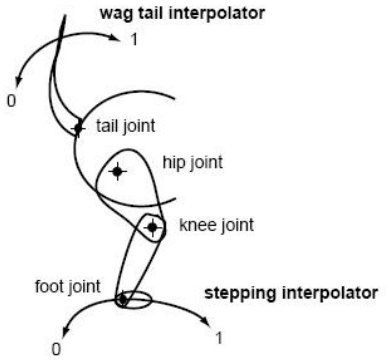

Figure 17: *Inverse kinematics is used to move the foot. A value between a 0 and 1 sets the leg along the stepping cycle, similarly for the wagging of the tail (Blumberg and Galyean [BG95])*

controlling simulated human and animal motions [vdP98, vdP97]. An important point illustrated is that, in order to build a successful system, there needs to be a way of evaluating the choices used to represent control functions. Analysis of real animals such as that provided by Alexander [Ale84], provides many insights into the movements of animals and the parameters that control them. However, this does not necessarily provide enough information to build a working controller.

In [vdP96] van de Panne uses parameters to synthesise different motions. By studying the motions produced, for example by looking at the speed with which a creature moves with different parameters, the optimal values can be found. Similarly, in order to animate a character, each has a starting pose and a desired pose. The torques needed to move the limbs from one pose to the next are calculated. Different trials are run to see which values for limb position and orientation give the quickest and most accurate result. Further enhancements can be used to speed up the process, such as using different levels of detail, as in the work of Torkos [Tor97], where the cat is simulated using physics for the overall body movement and IK for the legs. Whichever technique is used to help control the movement of a character, the end result needs to be a smooth realistic motion in real time. There is usually a compromise between the required level of control over the motion and the effort required to specify the motion.

4.3. Interaction and behaviour

Realistic behaviour is important for compelling quadruped simulation. Even if the generated gait is perfect, if the animal does not behave characteristically then its motion may be perceived to be anomalous.

Blumberg and Galyean [BG95] explored these issues by creating an interactive environment with virtual characters. They describe a system architecture that is able to propagate high level instructions to the geometry of a character, resulting in an animal which responds to human commands. The Behaviour system deals with instructions such as “find food” or “sit down and shake” as well as more simple commands like “move to” and “avoid obstacle”. These commands are translated into motor commands such as “forward”, “left”, “sit”. The Motor system then uses these instructions to produce coordinated motion such as walking, which in turn deforms the geometry of the character. This architecture is applicable to all creatures, as commands such as “move to” can be used for animals, humans or even dynamic objects such as cars. A controller ensures that “move to” is interpreted correctly by each creature. The movement of the limbs is controlled by inverse kinematics. As can be seen in Fig. 17 a value between 0 and 1 controls the position of the foot along the curved line, and the same technique is used to wag the tail. The behaviour of the actor depends on its relationship with the environment, whether a task is achievable or not, and also the priorities of actions.

Using artificial intelligence including action selection, machine learning, motor control and multi-agent coordination, Tomlinson and Blumberg [TB03] were able to bring their characters to life, displaying the same characteristics found in real wolves. They combine hard-coded behaviours (walking) with learned behaviours (being able to react to the environment), resulting in wolves that “grow” into different personalities as time goes on. The wolves are able to store previous interactions in their memory which can lead to new emergent behaviours. The wolves will learn from past experiences how to interact with other wolves in the pack, e.g, whether they are the dominant figure or not, and interpret other social cues. Users can also interact and control the virtual wolves by howling, barking, growling and whining into a microphone. By studying the social interactions between animals, a lot can be learned about behaviour within a pack. These actions can be broken down and even applied to human characters. This is especially useful in environments where autonomous behaviour is required. The desire to bring characters to life dates back as far as the first Disney cartoons, only today the demands are much greater. People are no longer satisfied with sitting back and watching a feature animation of hand animated characters telling a story. Today, interaction with characters in different and new ways is growing in popularity. The challenges to be met include creating flawless motion for any movement imagined and providing a way for the characters to think. Using animal characters as a testing ground for autonomous behaviours is appropriate since in real life they are independent beings that listen on occasion..

5. Conclusion

In this report we have discussed the rich and varied history of quadruped synthesis ranging from 2D cartoons to 3D anatomical representations. In all methods reviewed, the representation of animal motion in such a way as to be easily understood and controlled has been the main challenge. Even if humans are not overly familiar with the details of quadruped motion, we can still easliy pick out anomalies in computer generated simulations.

The most accessible sources of information today come from video footage. As we discussed, the extraction of motion from video is a very active topic in computer vision and has also proved very useful in computer animation. Animating characters through the use of skeletons is the most intuitive approach, but it is difficult to create realistic surface animations using such methods. Even for very nice skeletal motion the skin of a character can look incorrect if the angles of joints are too big or small, thus detracting from the realism of the motion. To counteract this, mesh animation has been developed, which in a sense uses the idea of a skeleton in order to position the vertices of the skin in the right place. However it is harder to animate such creatures as joints and bones are not explicitly represented. Hybrid methods involve the anatomical synthesis of animals, but this is hard to implement in real time, especially if the aim is to create a large herd of animals. The control of the character is ultimately what enables us to move them. With improved interfaces and effective representations of animal movements, it should be possible to create realistic motions very quickly.

Physically-based models are particularly interesting because they enable the generation of not only a single, but a full familly of plausible motions based on some a priori knowledge of the animal's morphology and on the way it uses its body to locomote. Requiring adapted control mecanisms for controlling the gait while maintaining balance, these techniques are still very challenging to develop. While generating a variety of motions by running a given model on different terrains has already been achieved, there is still room for research in the automatic generation of locomotion controllers for different animals, for instance based on the passive dynamics of their body [KRFC07].

Easing the task of creating compelling characters means that animators can spend their time directing scenes rather than tediously creating the movements for each individual character along with any resulting secondary motions. Other challenges involve creating different motions for similar characters, so that crowds or herds look like they are made up of individuals rather than clones. Parameterizing motion is a good way of achieving this goal, whereby different stylizations can be added to gaits.

References

[AHLD07] AUJAY G., HÉTROY F., LAZARUS F., DEPRAZ C.: Harmonic skeleton for realistic character animation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 151–160.

[AHSS04] AGARWALA A., HERTZMANN A., SALESIN D. H., SEITZ S. M.: Keyframe-based tracking for rotoscoping and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 584–591.

[AJ83] ALEXANDER R. M., JAYES A.: A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of Zoology 201* (1983), 135–152.

[Ale68] ALEXANDER R.: *Animal Mechanics*. Sidgwick & Jackson, 1968.

[Ale84] ALEXANDER R.: The gaits of bipedal and quadrupedal animals. In *The International Journal of Robotics Research* (1984), vol. 3, SAGE Publications, pp. 49–59.

[Ale96] ALEXANDER R.: *Optima for Animals*. Princeton University Press, 1996.

[Ale03] ALEXANDER R.: *Principles of Locomotion*. Princeton University Press, 2003.

[BG95] BLUMBERG B. M., GALYEAN T. A.: Multi-level direction of autonomous creatures for real-time virtual environments. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM, pp. 47–54.

[BK05] BARANOSKI G., KRISHNASWAMY A.: Simulation of light interaction with human skin. In *Eurographics 2005, Tutorial notes* (2005).

[Bos] Boston dynamics: www.bostondynamics.com.

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Trans. Graph. (Proceedings of SIGGRAPH) 26*, 3 (2007), 72.

[BPB06] BURN J., PORTUS B., BROCKINGTON C.: The effect of speed and gradient on hyperextension of the equine carpus. 169–171.

[CCC*05] CALIC J., CAMPBELL N., CALWAY A., MIRMEHDI M., BURGHARDT T., HANNUNA S., KONG C., PORTER S., CANAGARAJAH N., BULL D.: Towards intelligent content based retrieval of wildlife videos. In *Proceedings of the 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005)* (April 2005).

[CDNDMN85] CHARLES D. NEWTON D.V.M. M., DAVID M. NUNAMAKER V.: *Textbook of Small Animal*

Orthopedics. J.B. Lippincott Company, 1985. Normal and Abnormal Gait (Ch.91).

[Cla97] CLAYTON H.: Biomechanics of the horse. Presented by Dr. Hilary Clayton at the United States Dressage Federation Convention, December 6, 1997.

[Cla98] CLAYTON H.: Gait analysis: Past, present and future. The McPhail Chair Lecture given by Dr. Clayton at the annual Michigan Veterinary Conference on January 23rd, 1998.

[Cla01] CLAYTON H.: Gaits and movements of dressage horses. Presentations by Dr. Clayton at the Tufts Animal Expo. Boston, Massachusetts. 2001.

[dATTS08] DE AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.-P.: Automated conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum (Proc. Eurographics) 27*, 2 (2008), To appear.

[DSP06] DER K. G., SUMNER R. W., POPOVIC J.: Inverse kinematics for reduced deformable models. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 1174–1179.

[EKB95] EVANGELOS KOKKEVIS D. M., BADLER N. I.: Autonomous animation and control of four-legged animals. In *Graphics Interface '95* (1995), ACM Press / ACM SIGGRAPH, pp. 10–17.

[EM57] EADWEARD MUYBRIDGE L. S. B.: *Animals in Motion*. Dover Publications, 1957.

[FRDC04] FAVREAU L., REVERET L., DEPRAZ C., CANI M.-P.: Animal gaits from video. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 277–286.

[FTF03] FRANCIK J., TRYBICKA-FRANCIK K.: A framework for program control of animation of human and animal characters. *Studia Informatica, Silesian University of Technology 24*, 4 (2003), 55–65.

[GAD*07] GLUCKMAN P., ARETOS G., DOEPP D., PETERSON S., WALTMAN J., MODESTO L., CUTLER L., SENESHEN B.: From "shrek" to "shrek the third": evolution of cg characters in the "shrek" films. *ACM SIGGRAPH courses* (2007).

[GCT03] GIBSON D., CAMPBELL N., THOMAS B.: Quadruped gait analysis using sparse motion information. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (2003), vol. 3, pp. 333–6.

[Gir87] GIRARD M.: Interactive design of 3-d computer-animated legged animal motion. In *SI3D '86: Proceedings of the 1986 workshop on Interactive 3D graphics* (New York, NY, USA, 1987), ACM Press, pp. 131–150.

[GLH04] GOLUBOVIC D., LI B., HU H.: A hybrid software platform for sony aibo robots. 478–486.

[GMC*05] GLUCKMAN P., MINTER D., CHRONKHITE K., CURTIS C., HUANG M., VOGT R., SINGER S.: Madagascar: bringing a new visual style to the screen. *ACM SIGGRAPH Courses* (2005).

[GODC05] GIBSON D., OZIEM D., DALTON C., CAMPBELL N.: Capture and synthesis of insect motion. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 39–48.

[HCG05] HANNUNA S., CAMPBELL N., GIBSON D.: Identifying quadruped gait in wildlife video. In *International Conference on Image Processing* (September 2005), IEEE, pp. 713–716.

[HDK*06] HIEBERT B., DAVE J., KIM T.-Y., NEULANDER I., RIJPKEMA H., TELFORD W.: The chronicles of narnia: the lion, the crowds and rhythm and hues. *ACM SIGGRAPH Courses* (2006).

[ICE] Proceedings of the international conference on equine locomotion. Elsevier.

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. *ACM Trans. Graph. 24*, 3 (2005), 399–407.

[KRFC07] KRY P., REVERET L., FAURE F., CANI M.-P.: Modal locomotion: Controlling passive elastic dynamics. In *Siggraph Technical Sketches and Applications*. ACM Press, 2007.

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. (Proceedings of SIGGRAPH) 22*, 3 (2003), 954–961.

[Kuh03] KUHNEL J.: *Rigging a Horse and Rider: Simulating Predictable and Repetitive Movement of the Rider*. Master's thesis, Texas A & M University, 2003.

[Law] LAWSON S. E.: Equine mechanics: www.equinemechanics.com.

[LKA06] LIEN J.-M., KEYSER J., AMATO N.: Simultaneous shape decomposition and skeletonization. In *ACM Symposium on Solid and Physical Modeling* (2006), pp. 219–228.

[LvdPF00] LASZLO J., VAN DE PANNE M., FIUME E.: Interactive control for physically-based animation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 201–208.

[MDMT*04] MOCCOZET L., DELLAS F., MAGNENAT-THALMANN N., BIASOTTI S., MORTARA M., FALCIDIENO B., MIN P., VELTKAMP R.: Animatable human body model reconstruction from 3d scan data using templates. In *Workshop on Modeling and Motion Capture Techniques for Virtual Environments (CAPTECH'04)* (2004), pp. 73–79.

[Mil98] MILNE M.: Entertaining the future: Real animators don't rotoscope, or do they? *SIGGRAPH Comput. Graph. 32*, 2 (1998), 5–7.

[ML05] MARSLAND S., LAPEER R.: Physics-based animation of a trotting horse in a virtual environment. In *Proceedings of the 9th International Conference on Information Visualisation (IV'05)* (2005), pp. 398–403.

[RF03] RAMANAN D., FORSYTH D. A.: Using temporal coherence to build models of animals. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), IEEE Computer Society, p. 338.

[RFDC05] REVERET L., FAVREAU L., DEPRAZ C., CANI M.-P.: Morphable model of quadrupeds skeletons for animating 3d animals. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM, pp. 135–142.

[RH91a] RAIBERT M. H., HODGINS J. K.: Animation of dynamic legged locomotion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1991), ACM Press, pp. 349–358.

[RH91b] RAIBERT M. H., HODGINS J. K.: On the run. *ACM SIGGRAPH Electronic Theater* (1991).

[Sil04] SILVERMAN M.: *History of Rotoscoping*. Phoenix Editorial: http://www.measurand.com/motion-capture-resources/motion-capture-history-rotoscope.htm, 2004.

[SMT03] SEO H., MAGNENAT-THALMANN N.: An automatic modeling of human bodies from sizing parameters. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics* (2003), pp. 19–26.

[Stu94] STURMAN D. J.: Character motion systems: A brief history of motion capture for computer character animation. *ACM SIGGRAPH Courses* (1994).

[SWG02] SIMMONS M., WILHELMS J., GELDER A. V.: Model-based reconstruction for creature animation. In *ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), ACM Press, pp. 139–146.

[SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. In *ACM SIGGRAPH/Eurographics Symposium on Geometry Processing* (2007), pp. 153–162.

[SZT*07] SHI X., ZHOU K., TONG Y., DESBRUN M., BAO H., GUO B.: Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM Press, p. 81.

[TB03] TOMLINSON B., BLUMBERG B.: Alphawolf: Social learning, emotion and development in autonomous virtual agents. *Lecture Notes in Computer Science 2564* (2003), 35–45.

[TJ95] THOMAS F., JOHNSTON. O.: *The illusion of life : Disney animation*. New York : Hyperion, [1995], 1995.

[Tor97] TORKOS N.: *Footprint Based Quadruped Animation*. Master's thesis, University of Toronto, 1997.

[TP91] TURK M., PENTLAND A.: Eigen faces for recognition. *Journal of Cognitive Neuroscience 3*, 1 (1991), 71–86.

[TT04] THALMANN N. M., THALMANN D.: An overview of virtual humans. In *Handbook of Virtual Humans*. John Wiley, 2004, pp. 1–25.

[TvdP98] TORKOS N., VAN DE PANNE M.: Footprint-based quadruped motion synthesis. In *Proceedings of Graphics Interface '98* (1998), pp. 151–160.

[vdP96] VAN DE PANNE M.: Parameterized gait synthesis. In *Computer Graphics and Applications, IEEE* (1996), vol. 16, pp. 40–49.

[vdP97] VAN DE PANNE M.: Making them move: Motor control for animated humans and animals. In *European Control Conference* (1997), pp. 191–210.

[vdP98] VAN DE PANNE M.: Control for simulated human and animal motion. In *Atelier IFAC International Workshop on Motion Control* (1998), pp. 425–435.

[WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S. R., CANI M.-P., LIN M.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 13*, 2 (Mar-Apr 2007), 213–34. To appear.

[WFM01] WALTER M., FOURNIER A., MENEVAUX D.: Integrating shape and pattern in mammalian models. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 317–326.

[WG97] WILHELMS J., GELDER A. V.: Anatomically based modeling. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 173–180.

[Wid] WIDGERY R.: Kinetic impulse, horse locomotion: www.kinetic-impulse.com.

[WML*06] WU F.-C., MA W.-C., LIANG R.-H., CHEN B.-Y., OUHYOUNG M.: Domain connected graph: the skeleton of a closed 3d shape for animation. *The Visual Computer 22*, 2 (2006), 117–135.

[WP02] WADE L., PARENT R.: Automated generation of control skeletons for use in animation. *The Visual Computer 18*, 2 (2002), 97–110.

[WVG03] WILHELMS J., VAN GELDER A.: Combining vision and computer graphics for video motion capture. *The Visual Computer 19*, 6 (October 2003), 360–376.